Concordia University
Department of Computer Science and Software Engineering

SOEN422 Fall 2020
Lab B
Introduction to Arduino and Bare-Metal

Purpose

The goal of this lab is to allow student to allow students an opportunity to practice development on the AVR platform. For this lab exercise, use of the Arduino IDE is allowed and required but the use of the framework (functions, predefined objects ...etc) may not depending on the exercise. For this submission, submit your report as a PDF detailing your implementation, logic behind it and any other necessary discussion, as well as the code produced. The code for each task should be in separate files. You may submit your code as `.ino` files. Submissions must be in a `.zip` archive.

Task 1: Building a Logical Inverter

The first task is to implement a logical inverter. The output is the onboard LED on the Arduino, connected to `PB5` (Arduino pin 13). The input is to be PD6 (Arduino pin 6). Configure this pin as an input with its internal pull-up resistor enabled. **Use the Arduino Framework for this task.**

Task 2: Building a Logical Inverter (Again)

This task is the same as the first, except the implementation must be done in **raw C/C++**. In your report, discuss different implementations at both the hardware level and software level for this task.

Task 3: An Approach Sensor

Use the provided distance sensor, connected to `A2`. Every 5 seconds, check the sensor to see if it detects something approximately 4cm away from it or further and print the ADC value to the Arduino Serial Monitor. If something is detected at that range, turn on the built in LED on pin `PB5` (Arduino pin 13). This task is to be done **without the Arduino Framework except for using the `Serial` object**.
Some hints:

- When enabling the ADC, the first read is garbage. It is wise to clear out the ADC by doing this read before entering the main program loop.
- Use AV $_{cc}$ for the ADC reference voltage.
- For the ADC to work properly, the prescaler must be set so that the ADC will operate between 50 and 200kHz. This implies that `ADPS2`, `ADPS1` and `ADPS0` must be set since the Arduino operates at 16MHz.

Task 4: Variable Intensity Lights

Use an external LED connected to `OC0B` (`PD5`) and have an LED change brightness intensities every 5 seconds. Having the LED off counts as a brightness level. Perform this task using `raw C/C++`.

Task 5: Interrupting Doing Nothing

For this task, **there is to be nothing in a `while(1)` loop**. Using `INT0` and the corresponding built-in pull-up resistor, turn on the onboard LED for 5 seconds if the pin is pulled low. Do not use delays for this exercise. Use `raw C/C++`.

Resources

Below are some useful resources:

- ATmega328P Datasheet
- Sharp IR Sensor Datasheet
- Arduino Serial Library Reference
- AVR `util/delay.h` Reference, don't forget to set `F_CPU` to `16000000UL`
- AVR `avr/interrupt.h` Reference